

## REVERSE ENGINEERING OF CLINICAL SOFTWARE AND TYPES OF HAZARD

LALITA M. LOKHANDE & N. V. KALYANKAR

Department of Computer Science, Yeshwant Mahavidyalaya, Nanded, Maharashtra, India

### ABSTRACT

This paper is based on the hazards of Reverse Engineering Software of the system's code, documentation and behavior to create system abstractions and design information. Reverse engineering processes were used as a part of a software development, an implementation of a better program and to upgrade the existing documentation. The experiments carried out by recovering the information and it will implemented, dealing with some incomplete part of information and reverse engineering. The different types of Hazards are found in reverse engineering of the software, in this paper we have studied only few types hazards of reverse engineering of the software. To achieve this functionality, we have come across the situations like, how to load selected employee's entire record to main screen. This software has multifunctional uses a property which keeps data of salary, attendance, blood analysis and other types clinical test data of patient.

**KEYWORDS:** Reverse Engineering, Types of Hazards, Codes of Software

### INTRODUCTION

The practice of software engineering practice tends to focus on the design and implementation of a software product without considering its lifetime[1], usually ten years or more. However, the major effort in software engineering organizations is spent after development on maintaining the systems to remove existing errors and to adapt them to changed requirements.

Furthermore, reverse engineering is the process of learning the design of an object by studying its implementation. The core of reverse engineering consists in deriving information from the available software artifacts and translating it into abstract representations more easily understandable by humans[2-3]. The benefits are maximal when it is supported by tools. Reverse engineering can extract design information from source code, but the abstraction level, the completeness of the documents, the degree to which tools and a human analyst work together (interactivity), and the directionality of the process are highly variable[4].

The completeness of a reverse engineering process refers to the level of details that is provided at an abstract level. In most cases, completeness decreases as the abstraction level increases. For example, when a source code listing is given, Reverse engineering process refers to the completeness level of details that is provided at an abstract level[6]. In most cases, abstraction level will increases and completeness decreases. For example, when a source code listing is given, a complete procedural design representation is relatively easy to develop. Completeness depends on the way how the reverse engineer performs the analysis. Interactivity refers the human integration with automated tools which he used to create an efficient process of engineering. As abstraction level increases, interactivity must increase or completeness will suffer.

### Related Work

Reverse engineering process is directionally one-way; all the information extracted from the source code is provided to the software engineer who can then use it during any maintenance activity[7-8]. In other case, where the

directionality of the reverse engineering process is two-way, then the information is fed to a reverse engineering tool which tries to regenerate the old program. The extract abstraction is the core activity of reverse engineering process. A reverse engineer has to evaluate the old undocumented source code and develop meaningful specification of the processing, applied user interfaces and database used by the program[9].

For newly developed systems the problem can be reduced by thoroughly documenting the system and maintaining the documentation together with the system. Ideally, the system documentation describes the product and the complete design, including its rationale. The reason for performing reverse engineering is to maintain its code. Therefore, it should not be focused on program understanding but on system maintenance instead. This should be done in a way that frees us from reverse engineering a system again and again because of modifications made to its code over time. Therefore, understanding the increasing importance of reverse engineering of software, we wish to study hazards in reverse engineering of software's. The role of reverse engineering in software evolution is evidently important[10]. However, in order to introduce this discipline into the software engineering process, some fundamental building blocks are needed. Such are the availability of tools, data exchange settlements and a reverse engineering process.

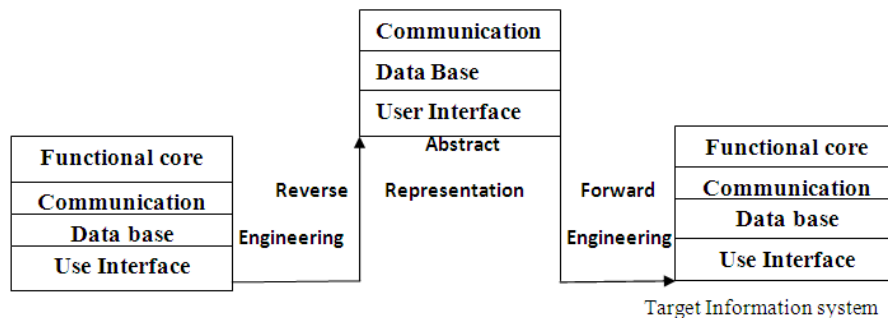
Chikosky and cross wrote in their research paper that the reverse engineering process applied to software means different things for different people[11]. They also define that 'reverse engineering is the process of analyzing a software system to create representations of the same system at higher level of abstraction. They also state that reverse engineering is the activity of going backward through the development process that means the output of the implementation phase which is in source code form is reverse engineered back to the analysis phase. It just like to go backward through the traditional waterfall model. Reverse engineering is the process of examination only as it does not modify the software. In practice two main types of reverse engineering situation may occur. In the first case, software's source code is available but with poor documentation. In other case, software's source code is not available. Any possible effort to discover one possible source code for the considered software is called as reverse engineering[12]. This second case is the one most people are familiar with.

Rather, a structure is superimposed in order to facilitate the execution of some software engineering task, e.g., maintenance or evolution. Structure imposition is Empire Software Engineering true also of design recovery, since in general multiple design views can be produced for a given system. The resulting structure may or may not be helpful depending on its end-users. Thus, the task-oriented nature of reverse engineering is necessarily complemented by a user-oriented characteristic. Moreover, usability issues are also central according to the definition above. In fact, the same piece of information recovered from the code may be immensely useful or completely unusable depending on the end user who is performing the current software engineering task and depending on the amount of knowledge the user already has about the system.

### **Using the Languages for Development of Software**

Reverse-engineering is now applied for software's, the process which was applied originally only to hardware, databases and even human Diribo nucleic acid. it is especially important with computer hardware and software. Programs were written in a language like C++, COBOL or Java, that's understandable by other programmers[13-18]. When programme will run on a computer, it had to be translated into another program, called a compiler, into the machine language. But machine code is difficult to understand by most programmers. So to convert this machine code back to human readable format a tool 'decompile' is used. Decompile is used for many purposes like as a learning tool, to make new more compatible and cheaper products than currently available in market. It also improves software's interoperability

and bridge data between different operating systems or database. Decompile is also used to discover the features of commercial product. How to do the reverse engineering of the software it is given below diagram[19-21].



**Figure 1: Reverse Engineering of Declarative User Interfaces**

### Types of Hazard Analysis

The main part of the System Safety effort is the hazard analysis. A hazard analysis is an examination of a system or subsystem to identify and classify each potential hazard according to its severity and likelihood of occurrence and to develop mitigation measures to those hazards to protect the public. A hazard analysis can take many forms. Typically, the following types of a hazard analysis are used. There are situations, called hazards, that prevent the next instruction in the instruction stream from being executing during its designated clock cycle[22]. Hazards reduce the performance from the ideal speedup gained by pipelining[23]. There are three classes of hazards:

#### Structural Hazards

It occurs when the hardware cannot support to the programme.

#### Data Hazards

It arises upon the previous instructions, when an instruction depends on the results in a way that is exposed by the overlapping of instructions.

#### Control Hazards

They arise from the pipelining of branches and other instructions that change the PC.

### Preliminary Hazard Analysis (PHA)

The PHA is begun early in the concept phase of the program or project to identify safety-critical areas, identify and evaluate hazards, and identify the safety design and operation requirements needed in the concept phase. The PHA provides the program or project with knowledge of the potential hazard causes and candidate controls. The PHA actually starts with a list of potential hazards, called a Preliminary Hazard List, to assist in the development of the PHA[24].

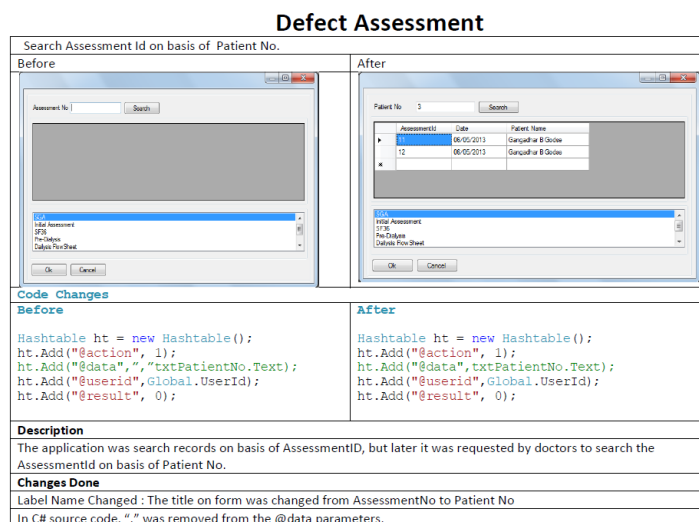
The general purpose of the SSHA is to perform a safety risk assessment of a system's subsystems at a greater depth than that provided in a PHA. The SSHA effort should begin when the preliminary design and concept definition are established, and it should continue through the detailed design of components, equipment, and software.

### Subsystem Hazard Analysis (SSHA)

This examines subsystems to determine how their: normal performance, Operational degradation, functional failure, unintended function and inadvertent function could contribute to system hazards. SSHA also determines how to satisfy design constraints on the subsystem design[25].

## Reengineering of Clinical Software

This software initially has major defects, hence it was not giving results properly. The major defects were removed by upgrading the source code as given belows. Now this clinical software is successfully used at various pathological laboratories. This software is multifunctional, which record attendance, data of the patient, in and out timing of workers and so on.



**Figure 2: Reserve Engineering of Dialysis Software by Tool Bar Hazards**

In application development we come across the situation where we would need a common Toolbar for all the form in application. Toolbars helps in having common interface for functionalities like Add, Edit, Delete and Search which is required on all the form.

Show Pre Dialysis reading with indicator, if the entered amount exceeds Male or Female Range.

### Code Changes

#### Before

Code are not given

#### After

```
string Range;

if (_patientObj.SexKey == "M")

Range=gvPredialysis.Rows[e.RowIndex].Cells["MaleRangeValue"].Value.ToString();

else

Range=gvPredialysis.Rows[e.RowIndex].Cells["FeMaleRangeValue"].Value.ToString();

Decimal ColumnValue = Convert.ToDecimal(e.FormattedValue);

if (Range.ToString().IndexOf("-") > -1)

{

if (Range.ToString().IndexOf("|") > -1)

{

//string[] RangeValues = Range.Split('-');
```

```

    }

    else

    {

        string[] RangeValues = Range.Split('-');

        decimal Value1 = Convert.ToDecimal(RangeValues[0]);

        decimal Value2 = Convert.ToDecimal(RangeValues[1]);

        if (ColumnValue < Value1 || ColumnValue > Value2)

        {

            //gvPredialysis[e.ColumnIndex,e.RowIndex].Style.Font= new Font("Verdana",10F, FontStyle.Bold);

            gvPredialysis[e.ColumnIndex,e.RowIndex].Style.BackColor= System.Drawing.Color.Yellow;

        }

        else

        {

            gvPredialysis[e.ColumnIndex,e.RowIndex].Style.BackColor= System.Drawing.Color.White;

        }

    }

    }

    if (Range.ToString().IndexOf(">") > -1)

    {

        string[] RangeValues = Range.Split('>');

        decimal Value2 = Convert.ToDecimal(RangeValues[1]);

        if (ColumnValue < Value2)

        {

            // gvPredialysis[e.ColumnIndex, e.RowIndex].Style.Font = new Font("Verdana", 10F,
FontStyle.Bold); gvPredialysis[e.ColumnIndex, e.RowIndex].Style.BackColor = System.Drawing.Color.Yellow;

        }

        else

        {

            gvPredialysis[e.ColumnIndex,e.RowIndex].Style.BackColor= System.Drawing.Color.White;

        }

    }

```

```

if (Range.ToString().IndexOf("<") > -1)
{
    string[] RangeValues = Range.Split('<');
    decimal Value2 = Convert.ToDecimal(RangeValues[1]);
    if (ColumnValue > Value2)
    {
        //gvPredialysis[e.ColumnIndex,e.RowIndex].Style.Font=newFont("Verdana",10F,FontStyle.Bold);
        gvPredialysis[e.ColumnIndex, e.RowIndex].Style.BackColor=system.Drawing.Color.Yellow;
    }
    else
    {
        gvPredialysis[e.ColumnIndex, e.RowIndex].Style.BackColor = System.Drawing.Color.White;
    }
}

```

### Description

In Pre Dialysis form, if the value entered in the Male or Female values are more than specified range, then the program should highlight the values entered in yellow color.

### Changes Done

In C# source code, the code was written in private void gvPredialysis\_CellValidating(object sender, DataGridViewCellValidatingEventArgs e) event. Which looks for values entered by users and if it find any values that is exceeding the limit specified than it highlights the value entered in Yellow color as shown in code.

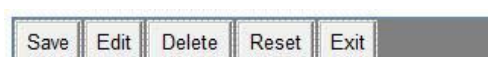


**Figure 3**

Implementing Toolbar is bit of task. People who wish to implement this feature need to know oops. Topics like overwriting, overloading and delegates are some which will be used in creating a Toolbar.

### How

To have Toolbar common for multiple form, we need to create a Toolbar using User Control. Drop 5 buttons and named them as Save, Edit, Delete, Reset and Exit on User Control.



**Figure 4**

In the code behind page, we have to create a EH and a delegate. After the USERCONTROL is ready, we have to implement this USER CONTROL now.

### Code

```
public partial class ucToolBar : System.Web.UI.UserControl
{
    public delegate void EventHandler(object sender, EventArgs e);
    public event EventHandler Save;
    public event EventHandler Edit;
    public event EventHandler Delete;
    public event EventHandler Reset;
    public event EventHandler Exit;
    protected void Page_Load(object sender, EventArgs e)
    {
    }
    protected void btnSave_Click(object sender, EventArgs e)
    {
        if (Save != null)
        {
            Save(sender, e);
        }
    }
    protected void btnEdit_Click(object sender, EventArgs e)
    {
        if (Edit != null) { Edit(sender, e);
        }
    }
    protected void btnDelete_Click(object sender, EventArgs e)
    {
        if (Delete != null) { Delete(sender, e);
        }
    }
}
```

```

protected void btnReset_Click(object sender, EventArgs e)
{
    if (Reset != null)
    {
        Reset(sender, e);
    }
}

protected void btnExit_Click(object sender, EventArgs e)
{
    if (Exit != null)
    {
        Exit(sender, e);
    }
}

```

Now we have confusion like where to implement USERCONTROL, whether on each form but this will kill the logic of Toolbar as will have to implement USERCONTROL on each page and if the application has 20 forms then implementing Toolbar on all pages will be a tedious job. So to overcome this issue we need a Base Page. Base Page will implement the Page object. Now the page that needs Toolbar will inherit Base Page. In Base page we have to use OnInit event.

### Code

```

public class BasePage : System.Web.UI.Page
{
    protected override void OnInit(EventArgs e)
    {
        ucToolBar uc = (ucToolBar)LoadControl("MyControls/ucToolBar.ascx");
        uc.Save += new ucToolBar.EventHandler(frmSave);
        uc.Edit += new ucToolBar.EventHandler(frmEdit);
        uc.Delete += new ucToolBar.EventHandler(frmDelete);
        uc.Reset += new ucToolBar.EventHandler(frmReset);
        uc.Exit += new ucToolBar.EventHandler(frmExit);
        this.FindControl("Form1").Controls.Add(uc);
    }
}

```



```

public virtual void frmSave(Object sender, EventArgs e) { }

public virtual void frmEdit(Object sender, EventArgs e) { }

public virtual void frmDelete(Object sender, EventArgs e) { }

public virtual void frmReset(Object sender, EventArgs e) { }

public virtual void frmExit(Object sender, EventArgs e) { }

}

```

This event is fired before any page event is fired. We can implement Toolbar in this event. A Toolbar to the menu on page object. To do this we have to S menu control on Derived Page. By doing this we managed to put the common Toolbar to our entire page. But there is issue of integration i.e. connecting derived page's event to click event on User Control. What happens is for example if we write code to save the data on form, it will be within the click event of the button. But what are trying to achieve is connect Save event with the click event of Save button on Toolbar. So for achieving this kind of connection we need the virtual methods on base page for each button. Attach this virtual method to the events of Toolbar implemented on OnInit event of Base Page.

#### Code

```

public virtual void frmSave(Object sender, EventArgs e) { }

public virtual void frmEdit(Object sender, EventArgs e) { }

public virtual void frmDelete(Object sender, EventArgs e){ }

public virtual void frmReset(Object sender, EventArgs e) { }

public virtual void frmExit(Object sender, EventArgs e) { }

}

```

#### Code

```

public override void frmSave(object sender, EventArgs e)

{

base.frmSave(sender, e);

Response.Write("Save");

}

public override void frmDelete(object sender, EventArgs e)

{ base.frmDelete(sender, e);

Response.Write("Delete");

}

public override void frmEdit(object sender, EventArgs e)

{

```

```

base.frmEdit(sender, e);

Response.Write("Edit");

}

public override void frmReset(object sender, EventArgs e)

{

base.frmReset(sender, e);

Response.Write("Reset");

}

```

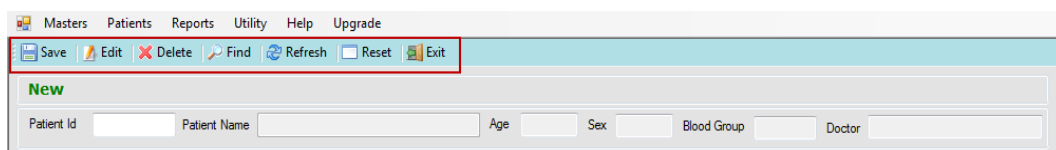


Figure 5: Out Put

Figure 6: Search Box Hazards

In application development, we come across the situations where there is a need of search screen which has option of searching the data based on some initial text entered. But to achieve this functionality, we come across the situations like, how to load selected employee's entire record to main screen. Question like whether to load selected employee's entire data on popup screen itself and send it to main screen in form of comma or pipe separated string or something else, comes to mind.

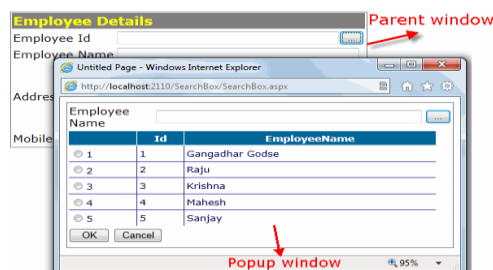


Figure 7

## How

To load selected employee's information on parent window, we need two small JavaScript functions, one on Parent Window page and other on Popup Window page. The JavaScript function on popup page will be called first as shown below.

## Code

```
function Select Data(Id)
```

```
{
    window.parent.CallLoadFunction(Id);
}
```

What I have done is, I called the Select Data() JavaScript function OK button's click event. This function will call the second JavaScript written on parent page and sends the selected employee's Employee Id to parent screen. On parent page, Call Load Function is called with Id as parameter as shown below.

#### Code

```
function Call Load Function(Id){
}
```

Here there is a small trick, I m using one hidden button because the button we have used on parent window is used for calling Search screen using button's On Client Click event. Now calling server side click event of a button will call it On Client Click event first and it will lead to more confusions, so I have decided to use hidden button and its click event to load selected employee's entire data on parent form.

A question might come to your mind is how are we going to load the server side event from JavaScript. The answer lies in two line code shown below, which is the complete JavaScript function that will be used on parent page.

#### Code

```
function CallLoadFunction(Id){
    var btnObject = document.getElementById('<% btnHidden.ClientID %>');
    btnObject.click();
}
```

First load the hidden button's object into javascript variable using Client ID property of button. And after the object is loaded into the btnObject, call its Click() event. On server side, i will write one event handler to handle hidden button's click event, which will populate employee details on parent page.

#### Code

```
protected void btnHidden_Click(object sender, EventArgs e)
{
}
```

So we have completed the coding all the sections, now it's time to test our search functionality.

## CONCLUSIONS

Reverse engineering improves the system structure, creates new system documentation and makes it easier to understand. Reverse engineering a software system has advantages over more radical approaches to system evolution. A hazard analysis type defines the analysis purpose, timing, scope, level of detail and system coverage; it does not specify how to perform the analysis. Several type of hazard analysis are found in system safety discipline that, together help ensure identification and resolution of system hazards. The reverse engineering of the software is used for better improvement and

proper function of software. The reserve engineering of the software of dialysis gives better and good results for the patient. This software is used for multifunctional properties which keep , salary, attendance, blood analysis and other types clinical test data of patient .

## ACKNOWLEDGEMENTS

Authors thank to UGC- Rajiv Gandhi National Fellowship for providing funds, Mr. Prashant Apte, Manager, e-consultancy of software's limited Mumbai , Dr. Rao, Department of Physics, Patkar College, Mumbai, Mr. Nilesh Kamble , Manager, Gargik technologies, Mumbai & Dr. Kalbande, HOD SPIT, Mumbai for providing necessary facilities and help for research work. Thanks to Yeshwant college management & Department of Computer Science for providing research facilities. Finally thanks to my family members, Dr. Manohar V. Lokhande, my daughter Mayuri and son Vishant.

## REFERENCES

1. Chih-Wei Lu, et.al, (1997) Reverse Engineering, Handbook of Software Engineering and Knowledge Engineering, Vol.2
2. Clayton. R, Rugaber. S., (1993) The Representation Problem In Reverse Engineering", Proceedings of the First Working Conference On Reverse Engineering, IEEE Computer Society Press, pp. 8-16.
3. Cohen, W.,(1995) Inductive Specification Recovery: Understanding Software by Learning from Example
4. Behaviors, Automated Software Engineering, Kluwer Academic Publishers, 2(2), 107-129.
5. Goldberg, B.E., (2007), System Engineering "Toolbox" for Design-Oriented Handbook of Software Engineering and Knowledge Engineering, Vol.2, Springer Science + Business Media, LLC Empire Software Eng DOI 10.1007/s10664-007-9037-5
6. Gerardo Canfora and Massimiliano DiPenta, New Frontiers of Reverse Engineering , (2007), Future of Software Engineering(FOSE'07)
7. Eric J. Byrne (1991) Software Practice and Experience, 21(12), 1349–1364
8. Muhammad Raza Ali (2005) Why Teach Reverse Engineering? ACM SIGSOFT Software Engineering Notes , 30(4)
9. Reverse engineering: a roadmap," in Proc. Conf. Future of Software Engineering, Limerick, Ireland, 2000, pp. 47-60.
10. G. Canfora and M. Di Penta, (2007), New Frontiers of Reverse Engineering, in Proc. Future of Software Engineering, Minneapolis, pp. 326-341.
11. Eric j . Byrne,(1991) Software reverse engineering: a case study, software—practice and experience, 21(12), 1349–1364
12. M. R. Ali, (2005) "Why teach reverse engineering?" ACM SIGSOFT SEN, 30(4), pp. 1-4.
13. V. Deursen, J. Favre, R. Koschke, and J. Rilling,(2003) "Experiences in Teaching Software Evolution and Program Comprehension," in Proc. 11th IEEE Int. Workshop on Program Comprehension, Washington, DC. pp. 2834-284.
14. E. Eliam, (2005) Secrets of Reverse Engineering, Indianapolis, IN: Wiley.

15. B. Gough(2005) introduction to GCCfor the GNU Compilers GCC and g++, Bristol, United Kingdom: Network Theory Limited.
16. L.Cunningham. (2008), COBOL Reborn [Online], Available: <http://it.toolbox.com/blogs/oracle-guide/cobol-reborn-25896>
17. B. W Weide, W D. Heym, J. E. Hollingsworth (1995), "Reverse engineering of legacy code exposed," in Proc. 17th Int. Conf. Software Engineering, Seattle, Washington, WA, pp. 327-331.
18. Wikipedia contributors. (2008). Compiler [Online]. Available:  
<http://en.wikipedia.org/w/index.php?title=Compiler&oldid=237244781>
19. Boomerang Decompiler Project. (2006), Boomerang: a general, open source, retarget able decompiler of machine code programs (Version 0.3.2) [Online]. Available: <http://boomerang.sourceforge.net>
20. K. Irvine(2007), Assembly Language: For Intel-Based Computers, Upper Saddle River, NJ: Prentice Hall.
21. Backer Street Software. (2007). REC: Reverse Engineering Compiler (Version 2.1) [Online]. Available: <http://www.backerstreet.com/rec/rec.htm>
22. Terry Hardy ,(2010), Hazard Analysis Types and Tools ,Great Circle Analytics, LLC
23. P.L.Clemens, W.T.Warner, (1995) A Perspective on System Safety,” Hazard Prevention, 1st Quarter, 1995.
24. C.A.Ericson,(2005) Hazard Analysis Techniques for System Safety, John Wiley & Sons.
25. B. E. Goldberg,(1994) System Engineering “Toolbox” for Design-Oriented.

